Integration framework for heterogeneous analysis components

Building a context aware virtual analyst

A. Bergeron-Guyard
DRDC – Valcartier Research Centre

Defence Research and Development Canada

Scientific Report
DRDC-RDDC-2014-R138
November 2014

Integration framework for heterogeneous analysis components

Building a context aware virtual analyst

A. Bergeron-Guyard
DRDC – Valcartier Research Centre

Defence Research and Development Canada

Scientific Report
DRDC-RDDC-2014-R138
November 2014

© Her Majesty the Queen in Right of Canada, as represented by the Minister of National Defence, 2014

© Sa Majesté la Reine (en droit du Canada), telle que représentée par le ministre de la Défense nationale, 2014

Abstract

Intelligence analysts are faced with information and cognitive overload problems. To address these problems, it is relevant to go beyond traditional knowledge exploitation and management approaches and make use of emerging cognitive support tools. This report proposes an integration framework to lay the groundwork for the creation of a context aware intelligence virtual analyst. A target integration framework architecture is proposed. An initial instantiation of analysis components on the proposed framework is also described.

Significance to defence and security

This effort lays the ground work and provides a way ahead for the development of an Intelligence Virtual Analyst Capability (iVAC). Exploiting the framework proposed in this document will enable the development of an Intelligent Software Assistant (ISA) that will support Canadian Armed Forces analysts in their collection, processing, analysis and dissemination tasks, thereby considerably reducing information and cognitive overload.

Résumé

Les analystes du renseignement sont aux prises avec des problèmes de surcharge informationnelle et cognitive. Pour régler ces problèmes, il est essentiel d'aller au-delà des approches traditionnelles de gestion et exploitation de la connaissance et d'utiliser des outils novateurs de support cognitifs. Ce rapport propose un cadre d'intégration pour jeter les bases d'un analyste virtuel du renseignement, sensible au contexte. Une architecture cible est présentée et une première version de composantes d'analyses déployées sur le cadre est décrite.

Importance pour la défense et la sécurité

Cet effort de recherche jette les bases et propose une direction pour le développement d'un *Intelligence Virtual Analyst Capability (iVAC).* L'utilisation du framework proposé dans ce document permettra le développement d'un *Intelligent Software Assistant (ISA)* qui aidera les analystes des forces armées canadiennes dans leurs tâches de collecte, traitement, analyse et dissémination en réduisant la surcharge cognitive et informationnelle.

Table of contents

List of figures

1 Introduction

The intelligence analysts of the Canadian Armed Forces have a mandate to collect, process and analyze information, and disseminate required intelligence. The main challenge facing the analysts is not a lack of data—in some way they are drowning in data—but rather managing and making sense of the large amount of data being presented to them. This overload problem (at the information and cognition levels) has recently been addressed using a variety of tools that allow extracting, analyzing, and reasoning on information [1]–[5].Still there remains a strong need to support the analysts, specifically in analyzing and making sense of the processed information in order to interpret its significance, and develop new knowledge.

In order to better address the overload problem, it is relevant to go beyond traditional knowledge exploitation and management approaches and make use of emerging cognitive support tools. A very promising paradigm in artificial intelligence has emerged: the Intelligent Software Assistant (ISA). The idea behind the research presented here is to use the ISA paradigm in the intelligence context and to synthesize the current state of artificial intelligence research in order to develop an Intelligence Virtual Analyst Capability (iVAC). An iVAC is a virtual analyst that organizes information, learns processes, adapts to changing situations, and interactively supports the analysts in their tasks in a seamless, intuitive fashion, eventually taking on autonomous tasks in concert with other analysts (virtual or human). An iVAC should be able to learn from its experience, by interacting with and being advised by its users. It should be able to explain what it is doing and why it is doing it. An iVAC should be aware of the context, such as traits and intent of its interactive "partners", and behave accordingly. An iVAC system should "be able to reflect on what goes wrong when an anomaly occurs, and anticipate such occurrences in the future. It should be able to reconfigure itself in response to contextual changes, and should be able to be configured, maintained, and operated by non-experts" [6].

The goal of this research is to propose an integration framework to lay the groundwork for the creation of a context aware intelligence virtual analyst. In order to put together a framework that allows for the incremental building of a virtual assistant, a flexible and scalable integration platform must first be proposed. A certain number of primary components are also required. Such primary components include a natural language processing capability, an avatar capability, and Graphical User Interface (GUI) capabilities.

Once the basic framework and preliminary components are available, context awareness and analysis components can be added to the system and deployed.

1.1 Organization of the document

This Scientific Report describes the proposed framework and its implementation at Defence Research and Development Canada (DRDC) – Valcartier Research Centre.

Section 2 presents the analysis that has been performed to identify the best technological and architectural solutions available for the integration framework and the primary components.

Section 3 presents the architecture that has been proposed for the integration framework.

Section 4 presents a first instantiation of a virtual analyst capability, based on the architecture proposed in Section 3. Primary components, as well as some preliminary context awareness and analysis components that have been developed and integrated, are presented.

2 Technological and architectural analysis

This section provides an overview of the analysis that was performed for the selection of each analysis component and for the selection of the framework. Explanations are meant to be succinct, very detailed information can be found at [7].

2.1 Analysis components

2.1.1 Natural Language Processing (NLP)

NLP is concerned with the interaction between humans and computer systems using human (natural) language. For the purpose of this work, NLP is being considered for three distinct aspects: Speech Recognition (SR), Natural Language Understanding (NLU), and Speech Synthesis (SS). Speech recognition is in charge of extracting written sentences out of the spoken (audio) input. Natural language understanding extracts meaning out of the provided sentences. Speech synthesis generates the audio output from sentences.

2.1.1.1 Speech recognition

Three of the principal avenues that have been investigated for speech recognition were: CMUSphinx, Julius, and Nuance Dragon.

CMUSphinx is an open source speech recognition system developed at Carnegie Mellon University. It contains several independent speech recognizers as well as Sphinxtrain, a set of acoustic model training tools. CMUSphinx can recognize speech, but does not contain a NLU Module. CMUSphinx possesses a Java and a C Application Programming Interface (API). The principal strength of CMUSphinx is its open source nature and a wide adoption in research labs and academia. It is highly customizable. The main drawbacks are a limited amount of training data used with the default acoustic model, and a lack of advanced speech recognition algorithms.

Julius is another open source speech recognition engine written in the C programming language. Both the dictation (which recognizes spoken utterances) and command and control (which understands commands) modes are supported. By default, Julius comes with the Japanese language support. English acoustic and language models are available from a third party for free, non-commercial use. Julius does not come with an advanced NLU capability. Julius is supposedly fast and efficient, but getting it effectively working with languages other than Japanese would require the investment of considerable resources.

Dragon Naturally Speaking is a speech recognition software package developed by Nuance Communications for the Windows operating system. Nuance also sells a Software Development Kit (SDK) allowing developers to create custom Windows applications with speech recognition (and synthesis) capabilities. The accuracy of speech recognition can be increased by adapting the software to each individual speaker’s accent and vocabulary. In the command and control mode, Dragon can recognize predefined sequences of words or patterns of words as commands. However, a separate NLU module is required to recognize more complex commands. Nuance’s speech recognition technology is widely accepted as the state-of-the-art. The main limitation is a relatively high cost.

Based on its performance, Dragon is recommended as the Speech Recognition component. The assumption here is that its acquisition cost would not surpass the customization cost of the other solutions.

2.1.1.2 Natural language understanding

A total of seven NLU libraries have been analysed. For the purpose of the project, an incremental solution has been proposed, leveraging different technologies. The incremental solution moves towards increasingly complex and powerful complementary NLU approaches: bag-of-words approach, hand-written grammar approach, and statistical NLP approach.

In its simplest approach, the NLU component would reduce the input sentence as a bag-of-words: a set of words contained in the sentence, removing the word ordering information. Meaning is attached to a given set.

The hand-written grammar approach consists in defining a context-free grammar for all acceptable queries. A parser is then used to derive the syntactic tree of input queries, from which the type and parameters of the task can be directly extracted.

Statistical NLP uses the full power of available statistical NLP taggers to extract a rich and flexible set of annotations, from which the task type and parameters should be extractable. This approach is more complex and closer to a full query understanding.

Implementing a bag-of-words approach is trivial, and does not require any sophisticated library. It only requires processing strings and lists, which is fully supported by any modern programming language. The main strength of this approach is its simplicity. Its main limitation is that the output of this approach is shallow and does not take into account the structure of the sentence, or query.

In order to implement the hand-written grammar approach, the use of PythonNLTK is recommended as it is the most convivial implementation of a hand-written, context free grammar parser.

For the statistical NLP approach, the ClearNLP library is proposed, as it covers most statistical NLP needs and is written in Java.

2.1.1.3 Speech synthesis

Ten solutions have been considered for SS, coming from both the open source and commercial communities. All reviewed open source speech synthesis technologies had one significant disadvantage: the sound of the synthesized voice was very artificial and sometimes difficult to recognize. This would be a serious limitation to intelligence analyst’s daily work. In terms of voice quality, Nuance Dragon proved to be the best alternative, and, as it is also being recommended for speech recognition, it becomes the recommendation for SS.

2.1.2 Avatar

Three of the principal avenues investigated for avatars were: Double Agent, Clippy.js and Guil3d.

Double Agent is an open source avatar technology from Microsoft that supports existing Microsoft Agent characters, including Microsoft Office Assistant characters. Double Agent relies on the discontinued Microsoft Agent software. The plugin is limited to specific versions of the Mozilla Firefox browser.

Clippy.js is a full JavaScript implementation of Microsoft Agent. Agents are composed of multiple image sequences that represent the frames of each animation related to an agent action. It is lightweight and easy to integrate. However, Clippy.JS does not include phonemes (Lip-Synching) animations. It does not allow for complex or refined avatar representations and is therefore limited.

Guil3D – Virtual Assistant Denise is a virtual assistant Windows desktop software with some artificial intelligence capabilities. Denise is a full feature application whose main function is to assist users in human-computer interaction. Denise can search the web, explore and play multimedia files, read and answer e-mails, schedule and remind appointments. Only desktop client integration is available in the current version. The software is also quite costly as more than $800 is required for a single user license of the enterprise version.

None of the surveyed avatar technologies meets the need of the iVAC. The current recommendation is therefore to develop a very simple avatar that will act as a place holder until appropriate avatar technology is made available or developed.

2.1.3 Graphical user interface

With regards to GUIs, the main alternatives are web-based or desktop GUIs. Web GUIs are designed to be run within a browser environment using various technologies to enable layout management, visual styling, as well as variable and object manipulation. The desktop alternative would require the installation of a client in order to drive the GUI. While each approach is valid and would have met the project's requirements, it has been decided to opt for a web-based approach for ease of integration within our existing legacy environment.

2.2 Framework

For the integration framework, the three considered alternatives have been: multi-tier distributed architecture, service integration, and service-oriented architecture.

A multi-tier architecture proposes a structure where presentation, business processing, and data management are logically separated. An application that uses middleware to service data requests between a user and a database employs a multi-tier architecture. The most widespread use of multi-tier architectures is the three-tier architecture.

A service integration approach involves the integration of applications through a service layer where services are aggregated, composed and consumed as needed. Applications are not directly connected to each other but interact through the service layer.

Service-Oriented Architecture (SOA) is a paradigm for the realization and maintenance of business processes that spam large distributed system. It is based on three major technical concepts: services, interoperability through an Enterprise Service Bus (ESB) and loose coupling. An ESB is used to implement communication between mutually interacting software applications. The ESB is in charge of monitoring messages between services, data transformation, mapping, and queuing.

For a full-fledged iVAC, the proposed approach would be a full service-oriented architecture. The use of the ESB would allow for the integration of complex heterogeneous components. However, for the initial context of this project, the service integration approach has been used. Service integration requires less effort for a first instantiation of the iVAC framework. Moreover, it is a steppingstone towards the SOA approach as the use of services will allow for a later transition to the full SOA using an ESB. The service integration architecture has been instantiated using the Java Enterprise Edition solution.

3 Integration framework

This section provides a description of the framework that was created to allow for the development of a virtual analyst. The details are kept at a more conceptual level; the actual technologies employed to instantiate the proposed model are secondary. The aim is to highlight the principles that allow for the integration of heterogeneous analysis components.

3.1 Query handling

Figure 1 provides an overview of the interaction between a user and the different parts of the system for the handling of queries.

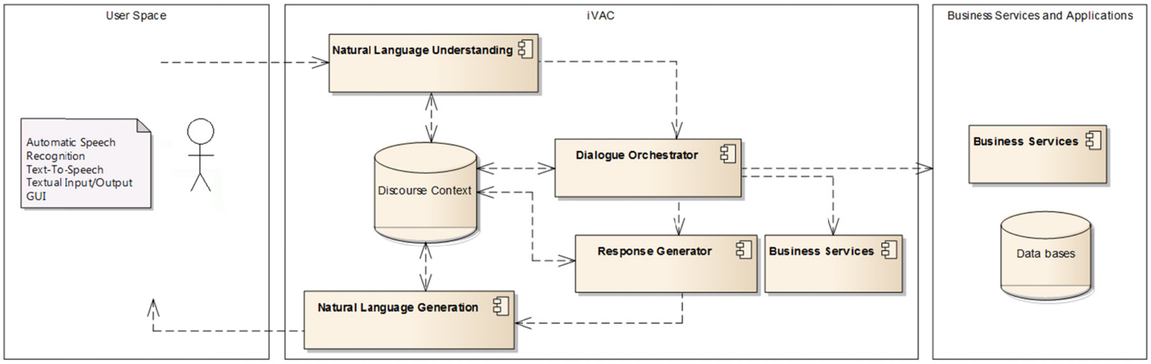


Figure 1: *High-level interaction for the handling of queries.*

The user will interact with the system using spoken or written natural language, or using the GUI. A natural language query will be handled by the Natural Language Understanding module. The Natural Language Understanding component translates the user voice into a command. This module will use information stored in the Discourse Context, which contains the data (bag of words, grammar, statistical analysis data) required to perform Natural Language Processing (details provided in Section 2.1.1.2). The Dialogue Orchestrator receives the user request and calls the necessary services (Business Services) to execute the command. The Response Generator produces a response from the received results. The Natural Language Generation of Figure 1 formulates the response in natural language.

3.2 Framework functions

Figure 2 shows a more detailed view of the framework functions. In some cases technologies are associated with particular functions. Although the specified technologies are indeed candidates of choice for the specified function, it must be understood that the function itself is essential to the system, while the technological components remain interchangeable.

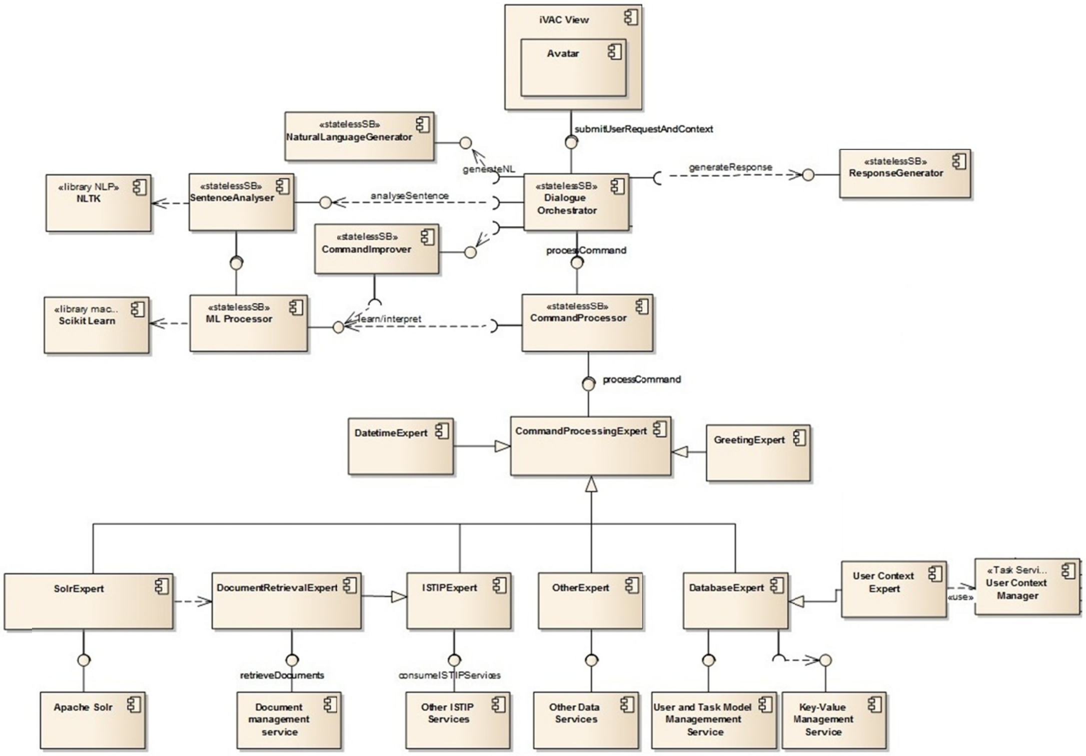


Figure 2: *Framework functions.*

At the top of Figure 2, the iVAC view is where the user interacts with the system (as detailed in Section 3.1). This interaction is made using a GUI, natural language processing (including speech synthesis and speech recognition), and an avatar representation.

3.2.1 Dialogue management

The following components are used for dialogue management:

- The **Dialogue orchestrator** coordinates the activities of all the components of the dialogue system;
- The **Sentence analyser** allows for the system to understand natural language user requests;
- The **Library NLP** is a third party library that supports the Sentence Analyser to process and understand the spoken request;
- The **Command improver** is called by the Dialogue Orchestrator when the latter needs to get a more accurate command in order to call core business functions correctly. The Command Improver uses a Machine Learning (ML) Processor;
- The **ML processor** provides machine learning capability to the communication system. Based on the user profile, behaviour and feedback, the system could learn to better understand natural language queries;
- The **ML library** (ML processor) is a third party library that supports the ML processor;
- The **Response generator** aggregates all the responses returned by the command processing experts; and

- The **Natural language generator** produces a textual utterance from the response delivered to the dialogue orchestrator.

3.2.2 Command processing

The following components are used for command processing:

- The **Command processor** is aware of all the business functions that are available to process specific commands. It maps and forwards commands to the appropriate processing service.
- The **Command processing expert** is an abstract command processing component. It provides a well-defined interface to specialized experts. By implementing this interface, it is possible to add new expert or analysis services to the system. For the initial version of the framework, the following expert services were implemented:
  - **ISTIPExpert:** handles tasks or commands related to services hosted in the DRDC developed Intelligence Science and Technology Integration Platform (ISTIP) [1] infrastructure. Depending on the type of the command, this component calls the right ISTIP service to process the command;
  - **DocumentExpert:** handles documents retrieval and/or processing commands. It relies on a document processing system or services to process the command;
  - **GreetingExpert:** greets the user and initialises the user session context;
  - **DateTimeExpert:** handles date and time-related commands;
  - **DataBaseExpert:** queries database systems. It issues data queries to its known databases. It relies on the interfaces and/or services provided by those database systems;
  - **UserContextExpert:** manages user context related tasks; and
  - **SolrExpert:** retrieves documents and/or similar user contexts in the Solr indexation system and by using Solr [18] advanced search capabilities.

3.3 Framework software architecture

This section provides an overview of the software components that have been used to implement the first version of the integration framework as shown on Figure 3. This overview of the technological stack is provided as a reference; additional details can be found at [8].

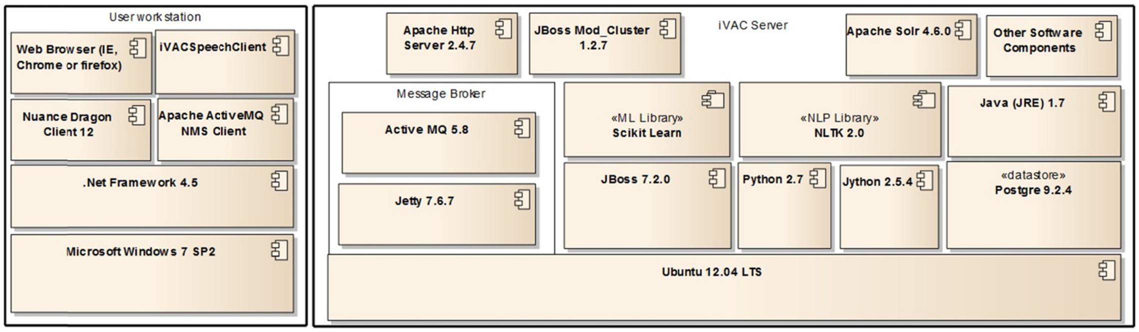


Figure 3: Software architecture.

The following software systems are used on the server side in the initial version of the integration framework:

- The **Machine Learning (ML) library:** Scikit Learn [9] is a software library that handles the learning capability of the iVAC;
- The **Natural Language Processing (NLP) library:** for the iVAC, the «NLTK 2.0» [10] is used as the natural language processing library of choice;
- **Python 2.7:** Python [11] is an interpreted language designed to speed up development time for rapid prototyping. The Python interpreter is required by the NLTK components;
- **Jython 2.5.4:** Jython [12] is Python for Java platforms. It is used to create and support a service adapter to integrate the NLU components;
- **Ubuntu 12.04 LTS:** Ubuntu [13] Server is a Unix-like operating system. It is used on the server side of the iVAC system;
- **JBoss 7.2.0:** JBoss 7 [14] is a fully certified Java enterprise Edition 6 server application. It is used to host the iVAC business component;
- **Apache Active MQ 5.8:** Active MQ [15] is an open source message broker and integration platform. It supports cross language messaging;
- **Jetty 9:** Jetty [16] is an open source Java Servlet Container. As such, it provides a web server and servlet hosting capability. It is the Web server that hosts the management console of Active MQ;
- **Java Runtime environment 1.7:** The Java Runtime environment [17] runs all Java-based programs. As such, environment software systems like JBoss, Active MQ and Tomcat or Jetty cannot be run without Java RE;
- **Apache Solr 4.6.0:** Solr [18] is an open source enterprise search platform from the Apache Lucene project. Its main features include powerful full-text search, hit highlighting, faceted search, near real-time indexing, dynamic clustering, database integration, rich document (e.g., Word, PDF) handling, and geospatial search; and
- **Apache HTTP server 2.4.6:** the Apache [19] HTTP Server coupled with the JBoss Mod_Cluster 1.2.7 [20] acts like a load balancer for the iVAC cluster. Its main purpose is to distribute the load between the nodes that form the iVAC Server Cluster.

The following software systems are used on the client side in the initial version of the integration framework:

- **Microsoft Windows 7 Professional:** Microsoft Windows 7 [21] is the client version of the Microsoft Windows operating system;
- **Microsoft .Net Framework 4.5:** .Net Framework [22] is an execution environment for Microsoft .Net components, tools and framework;
- **Nuance Dragon Client 12:** Nuance Dragon [23] is a software system with several capabilities; among them: Speech Recognition and Speech Synthesis; and
- **ActiveMQ NMS Client 2.0:** ActiveMQ NMS Client [24] is a .Net client that communicates with the ActiveMQ Message Broker.

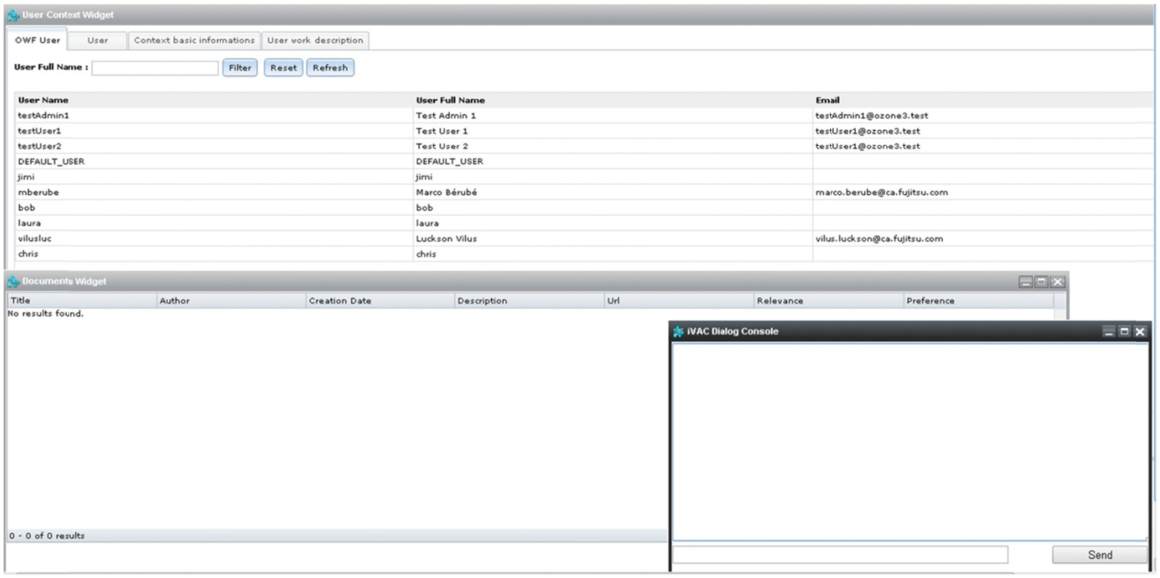
4 Initial framework instantiation

This section describes a first version of the framework that has been built according to the specifications detailed in Section 3. This framework contains a limited set of analysis components that mainly deal with user and context management, as well as with documents retrieval and recommendation.

The initial application deployed on the framework performs documents retrieval. The user is able to perform document searches based on keywords. The application is also able to take into account the user's context (identity, role and preference) and the user's feedback (on previously retrieved documents) to retrieve documents. Context-aware documents retrieval is the topic of Section 4.2. Section 4.1 first provides a description of the application.

4.1 Application description

Figure 4 shows the application dashboard.



User Name	User Full Name	Email
testAdmin1	Test Admin 1	testAdmin1@ozone3.test
testUser1	Test User 1	testUser1@ozone3.test
testUser2	Test User 2	testUser1@ozone3.test
DEFAULT_USER	DEFAULT_USER	
jimi	jimi	
mberube	Marco Bérubé	marco.berube@ca.fujitsu.com
bob	bob	
laura	laura	
vilusluc	Luckson Vilus	vilus.luckson@ca.fujitsu.com
chris	chris	

Title	Author	Creation Date	Description	Url	Relevance	Preference

No results found.

0 - 0 of 0 results

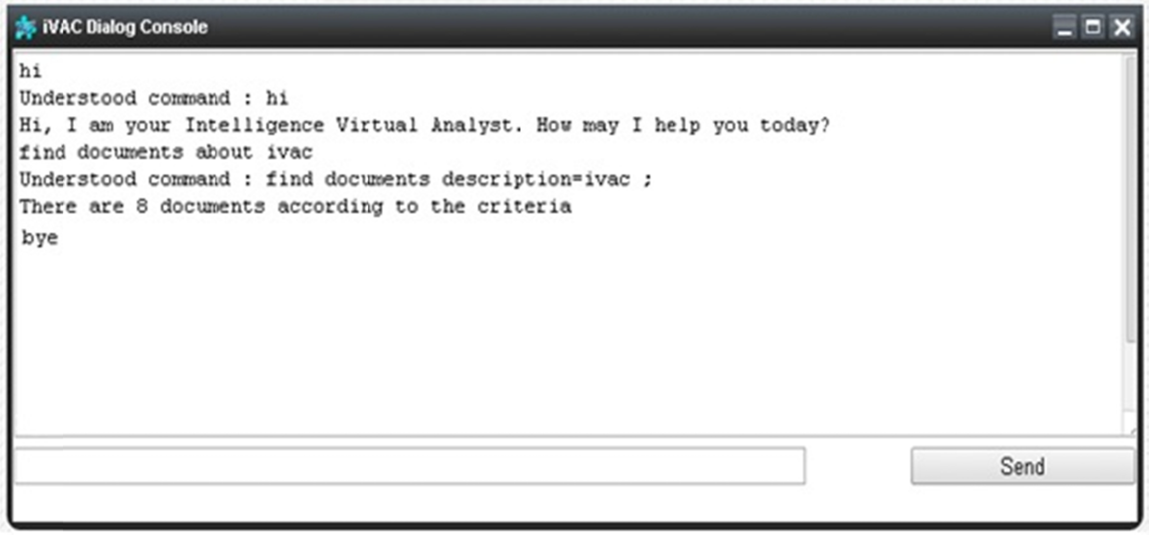
Figure 4: *Application dashboard.*

The dashboard contains three main sections:

- the dialogue console widget;
- the documents list widget; and
- the user context definition widget.

4.1.1 Dialogue console widget

Figure 5 shows the dialogue console widget.



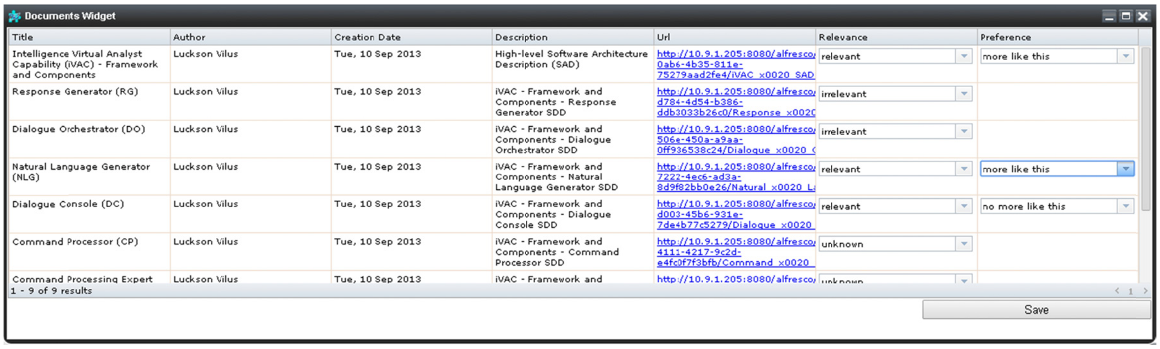
```
hi
Understood command : hi
Hi, I am your Intelligence Virtual Analyst. How may I help you today?
find documents about ivac
Understood command : find documents description=ivac ;
There are 8 documents according to the criteria
bye
```

Figure 5: *Dialogue console widget.*

The dialogue console widget is mainly used to submit user requests to the system. The user types his requests in natural language and hits the send button to send the request to the system. The system uses the bag of words and grammar approaches (Section 2.1.1.2) to extract a precise command from the written utterance. The understood query is given back to the user in the window.

4.1.2 Documents widget

Figure 6 shows the document widget.



Title	Author	Creation Date	Description	Url	Relevance	Preference
Intelligence Virtual Analyst Capability (iVAC) - Framework and Components	Luckson Vilus	Tue, 10 Sep 2013	High-level Software Architecture Description (SAD)	http://10.9.1.205:8080/alfresco/ 0ab6-4b35-811e- 75279aad2fe4/iVAC_x0020_SAD	relevant	more like this
Response Generator (RG)	Luckson Vilus	Tue, 10 Sep 2013	iVAC - Framework and Components - Response Generator SDD	http://10.9.1.205:8080/alfresco/ d784-4d54-b386- ddb3033b26c0/Response_x002C	irrelevant	
Dialogue Orchestrator (DO)	Luckson Vilus	Tue, 10 Sep 2013	iVAC - Framework and Components - Dialogue Orchestrator SDD	http://10.9.1.205:8080/alfresco/ 506e-450a-a9aa- 0ff936538c24/Dialogue_x0020_(	irrelevant	
Natural Language Generator (NLG)	Luckson Vilus	Tue, 10 Sep 2013	iVAC - Framework and Components - Natural Language Generator SDD	http://10.9.1.205:8080/alfresco/ 7222-4ec6-ad3a- 8d9f82bb0e26/Natural_x0020_La	relevant	more like this
Dialogue Console (DC)	Luckson Vilus	Tue, 10 Sep 2013	iVAC - Framework and Components - Dialogue Console SDD	http://10.9.1.205:8080/alfresco/ d003-45b6-931e- 7de4b77c5279/Dialogue_x0020_	relevant	no more like this
Command Processor (CP)	Luckson Vilus	Tue, 10 Sep 2013	iVAC - Framework and Components - Command Processor SDD	http://10.9.1.205:8080/alfresco/ 4111-4217-9c2d- e4fc0f7f3bfb/Command_x0020_	unknown	
Command Processing Expert	Luckson Vilus	Tue, 10 Sep 2013	iVAC - Framework and	http://10.9.1.205:8080/alfresco/	unknown	

1 - 9 of 9 results

Figure 6: *Document widget.*

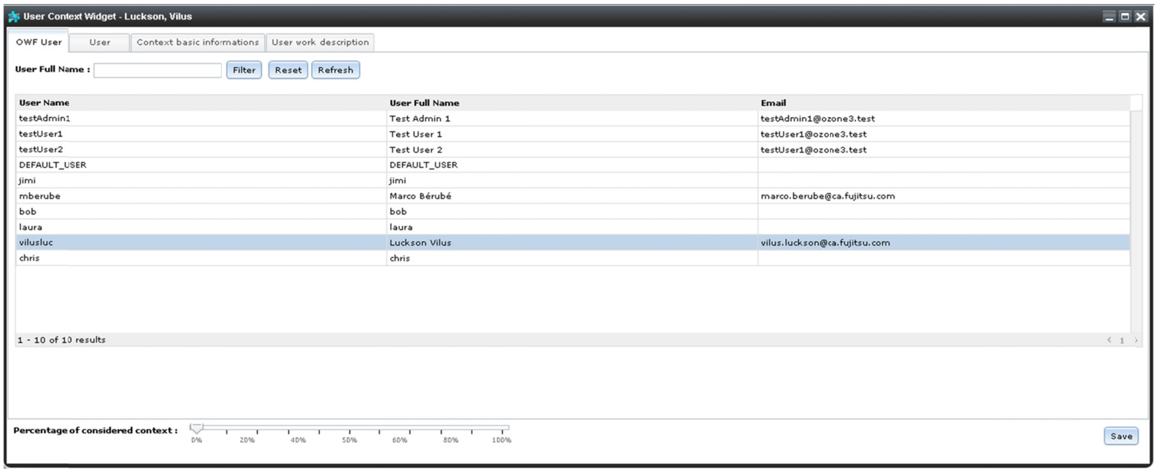
The documents widget is used to display a list of documents returned by the system for a submitted request. Using this widget, the user can also provide feedback to the system. To provide feedback, the user selects a document on the grid and then selects one of the following options in the relevance field:

- *Unknown*: the default option, which provides no actual feedback.
- *Relevant*: the user considers this document as relevant. When he selects this option, other feedback options become available. The user can select one of these options:
  - *More Like This*: the user wants to get more documents like this one; and
  - *No More like This*: even if the user considers this document as relevant, he does not want to retrieve other similar ones.
- *Irrelevant*: the document is not relevant for the current context of the user.

After making his choices, the user can hit the «save» button in order to submit his feedback to the system.

4.1.3 User context widget

Figure 7 shows the user context widget.



User Full Name :

User Name	User Full Name	Email
testAdmin1	Test Admin 1	testAdmin1@ozone3.test
testUser1	Test User 1	testUser1@ozone3.test
testUser2	Test User 2	testUser1@ozone3.test
DEFAULT_USER	DEFAULT_USER	
jimi	jimi	
mberube	Marco Bérubé	marco.berube@ca.fujitsu.com
bob	bob	
laura	laura	
vilusluc	Luckson Vilus	vilus.luckson@ca.fujitsu.com
chris	chris	

1 - 10 of 10 results

Percentage of considered context :

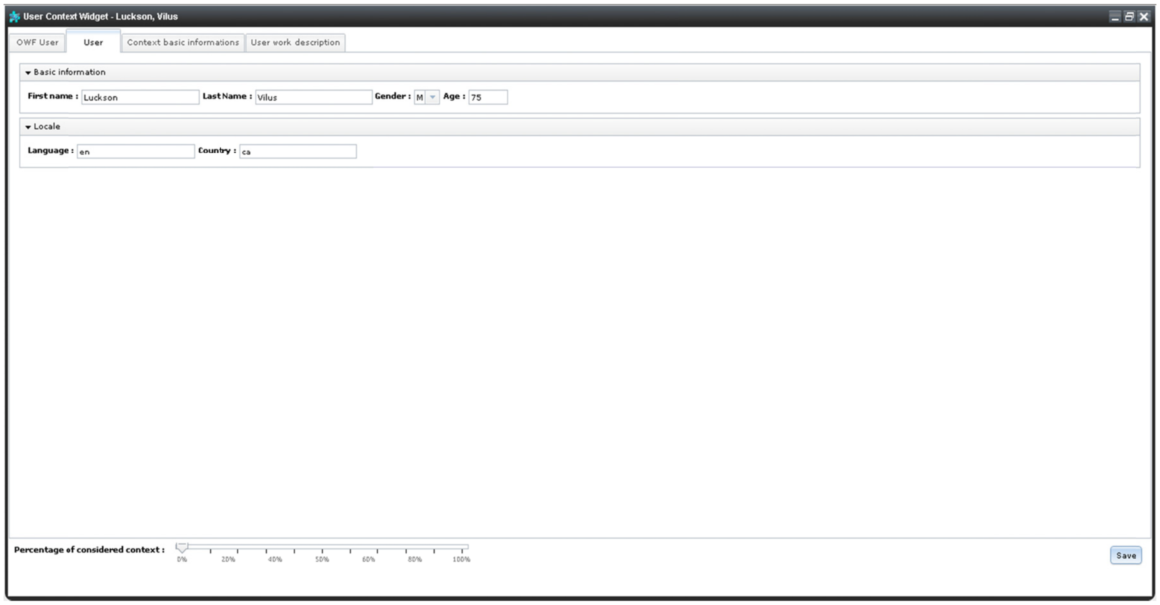
Figure 7: *User context widget—Ozone Widget Framework (OWF) user tab.*

The user context widget is used to define the context of a user. This widget contains four tabs:

- *OWF User* (shown on Figure 7): this grid displays the list of users contained in the Ozone Widget Framework database. This is accessible only by users having administrative privileges;
- *User* (shown on Figure 8): this contains basic information about the user;
- *Context basic information* (shown on Figure 9): this contains the information related to the context like:
  - A collection of keywords defined by the user; and

    - A collection of relevant documents.

- *User work description* (Figure 10): this is where the user role is described (his mission, his function and his tasks).



User Context Widget - Luckson, Vilus

Basic information

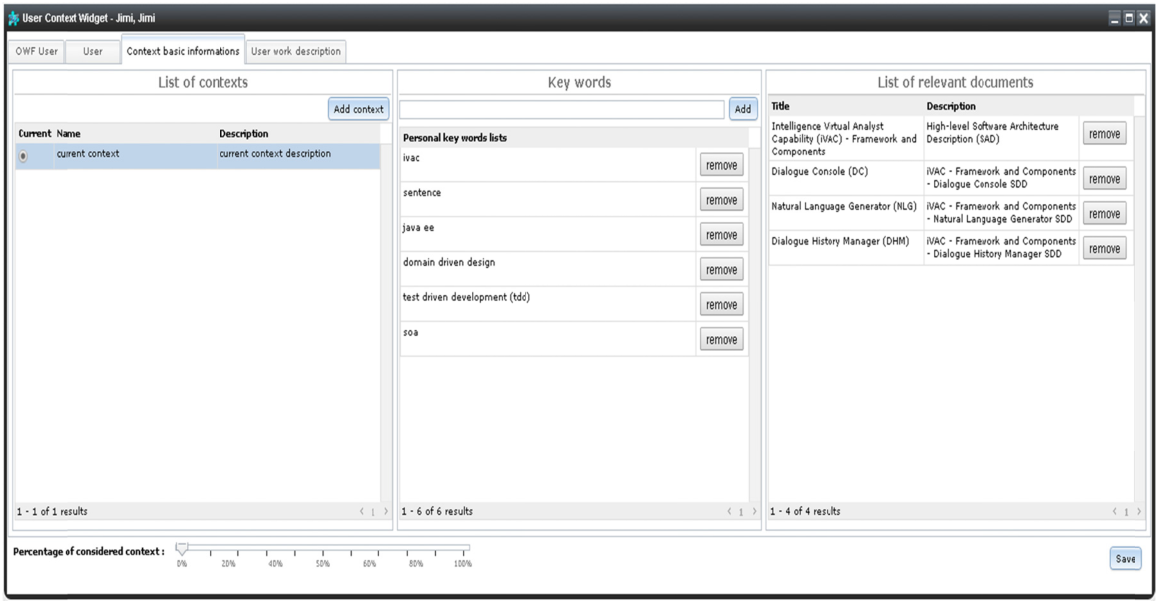
First name : Luckson Last Name : Vilus Gender : M Age : 75

Locale

Language : en Country : ca

Percentage of considered context : 0% 20% 40% 50% 60% 80% 100%

Figure 8: *User context widget—User tab.*



User Context Widget - Jimi, Jimi

List of contexts

Current	Name	Description
●	current context	current context description

1 - 1 of 1 results

Key words

Personal key words lists
ivac
sentence
java ee
domain driven design
test driven development (tdd)
soa

1 - 6 of 6 results

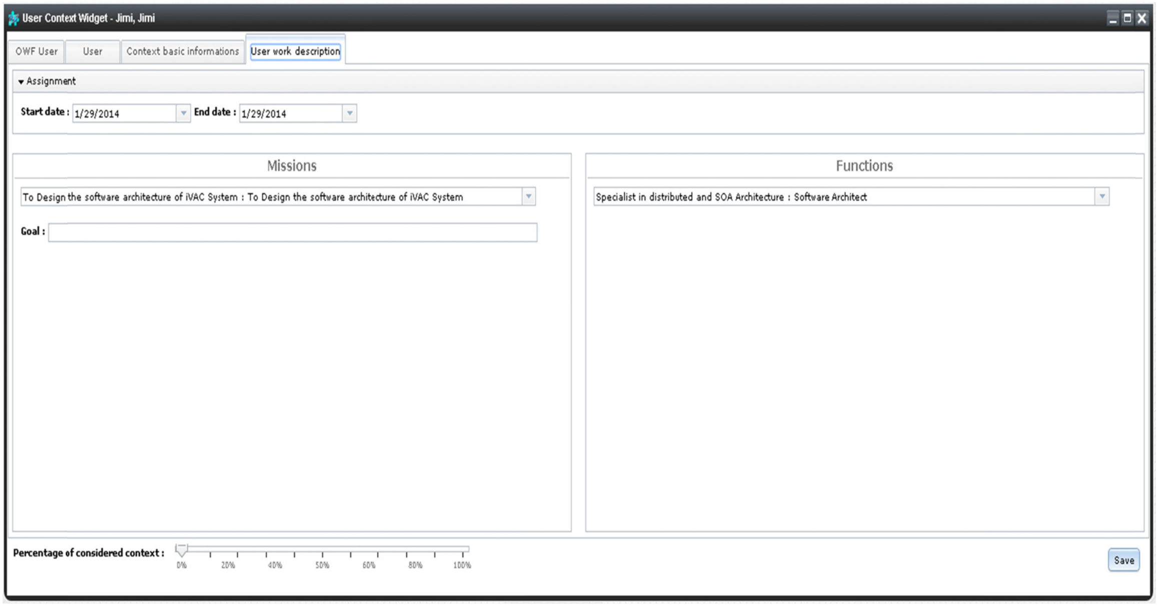
List of relevant documents

Title	Description
Intelligence Virtual Analyst Capability (iVAC) - Framework and Components	High-level Software Architecture Description (SAD)
Dialogue Console (DC)	iVAC - Framework and Components - Dialogue Console SDD
Natural Language Generator (NLG)	iVAC - Framework and Components - Natural Language Generator SDD
Dialogue History Manager (DHM)	iVAC - Framework and Components - Dialogue History Manager SDD

1 - 4 of 4 results

Percentage of considered context : 0% 20% 40% 50% 60% 80% 100%

Figure 9: *User context widget—Context basic information tab.*



Start date : 1/29/2014 End date : 1/29/2014

To Design the software architecture of iVAC System : To Design the software architecture of iVAC System

Goal :

Specialist in distributed and SOA Architecture : Software Architect

Percentage of considered context :

__Figure 10:__ User context widget—User work description tab.

Notice the “percentage of considered context” slider at the bottom of the figure, which allows for the user to move from a strictly keyword-based search (0%) to a strictly context-based search. This is the topic of the following section.

4.2 Context aware document retrieval

This section provides an overview of the context-aware document retrieval mechanics. Detailed information can be found in [25].

4.2.1 Keywords and context

Context-aware document retrieval aims at providing users with documents that are not only relevant to a particular keyword, but that also take into account the user’s context. Figure 11 illustrates this notion.

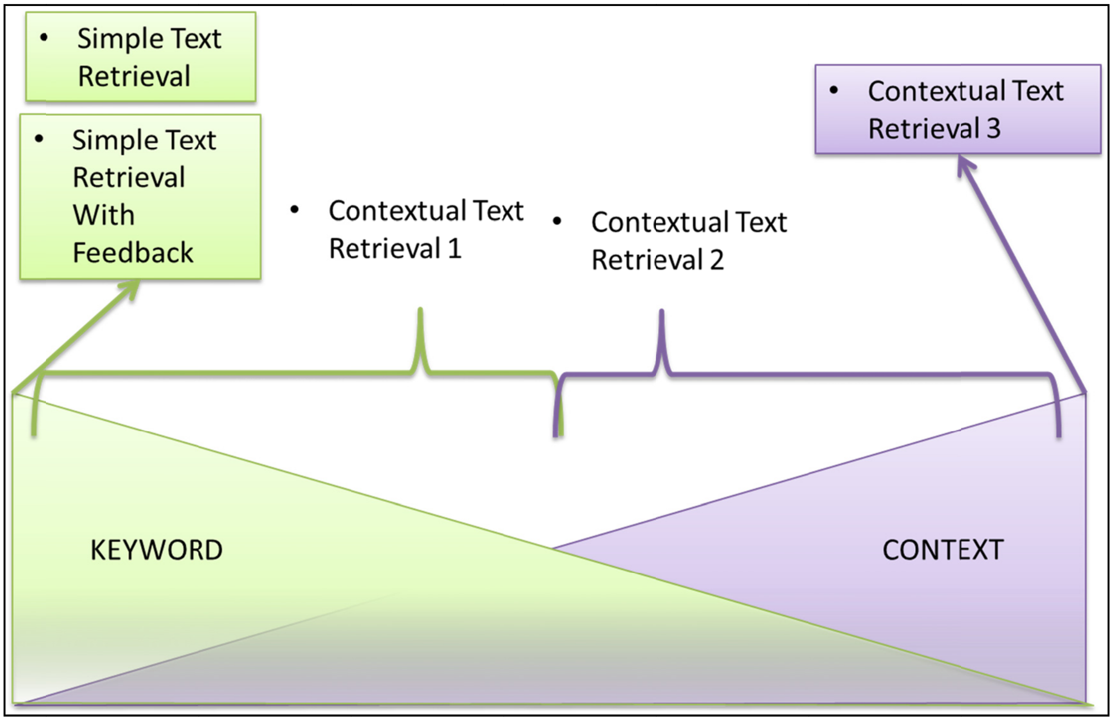

Simple Text Retrieval
Simple Text Retrieval With Feedback
Contextual Text Retrieval 1
Contextual Text Retrieval 2
Contextual Text Retrieval 3
KEYWORD
CONTEXT

Figure 11: *Context vs. keyword relative importance.*

At its simplest form (*Simple Text Retrieval*), the system will use only the keywords provided by the user. The system will behave the same way as any typical search engine would, by retrieving documents containing the keywords. The system is also able to consider user feedback (*Simple Text Retrieval With Feedback*) (e.g., relevant, irrelevant, more like this, no more like this—see Section 4.1.2) to refine search results. Finally, the system is able to consider the user's context to refine search results. Actually, the system can move from keyword-based to context-based retrieval using a user-specified keyword-context ratio (illustrated on Figure 10). As previously mentioned, when located at the left end of the spectrum (green of Figure 11, basic keyword search will be performed. When located at the right (purple of Figure 11), only the context of the user will be used. This means that the retrieval result will not take into account the provided keywords and will strictly contain context-based results.

Having such a flexible system, allowing to move from keyword-based to context-based retrieval, provides the user with various types of results, some of which might not have been made available by a standard document retriever. The idea behind this approach is that a user may, indeed, be looking for documents containing specific keywords. However, it is also possible that a user may be inputting keywords that are somewhat arbitrary in order to find documents of interest that are not related to (and may not contain) the specified keywords. In this case, context-based retrieval is a complimentary solution that is likely to return different documents of interest.

The following sections provide details on the approaches powering keyword-based retrieval, context-based retrieval, and user feedback.

4.2.2 Keyword-based document retrieval

To perform keyword-based searches, the system uses the Solr [18] search engine. Solr provides the following functionalities:

- Index documents;
- Return all the terms indexed;
- Return the term frequency, document frequency and Term Frequency-Inverse Document Frequency (TF-IDF) values for any indexed document;
- The possibility to weight the importance of keywords in a query;
- The possibility to search on multiple document fields; and
- The possibility to weight the importance of document fields in a query.

In Solr, documents are composed of fields, which are specific pieces of information. Fields can contain different types of data (e.g., date-time, binary, boolean, currency, Unicode).

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistic that reflects the importance of a word in a document. It is the combination of two measures: Term Frequency (TF) and Inverse Document Frequency (IDF). TF considers the frequency of a given term in a document. IDF is a way to measure the amount of information a given word provides by evaluating if it is common across all documents. Roughly, IDF is computed by dividing the total number of documents by the number of documents containing a given word. TF-IDF is produced by multiplying both measures. On a general document corpus, for any document, the term "the" would likely score a high TF measure (as it is frequently used), and very low IDF (as it is probably used in most documents). The combined TF-IDF measure for "the" would be low, reflecting the notion that the word "the" does not convey much importance in the meaning of a given document. Let's say a particular document in our general corpus discusses Neutrinos. The TF measure for "neutrino" in this specific document would likely be high (since it is the topic of the document). On the other hand, since the corpus is of a general nature, it is likely that few other documents would contain "neutrino". Therefore, the IDF measure would also be high. The combined TF-IDF measure for "neutrino" would be high, reflecting that this particular word is of special importance for the document at hand.

4.2.3 Handling user feedback

Relevance feedback is an approach that modifies the weights of keywords in the request based on the relevant documents identified by the user. This is implemented using a Rocchio Equation [26]. In general terms, the Rocchio Equation adds to the original weights of terms the average importance of the word in the relevant documents and subtracts the average importance of the word in the irrelevant documents. The importance of a given keyword is computed using the TF-IDF method described in 4.2.2.

In the context of the application, the user is allowed to specify feedback using Unknown, Relevant-More Like This, Relevant-No More Like This, or Irrelevant (see Section 4.1.2). In practice, the Unknown and Relevant-No More Like This options do nothing. The Relevant-More

Like This option is used to add importance to keywords present in the document. The Irrelevant option is used to reduce the importance of keywords present in the document.

4.2.4 Handling context

In this initial version of the system, the context is handled using the keywords and "relevant documents" identified by the user (see Section 4.1.3—Context Basic Information Tab). The system will simply use the provided keywords along with the identified relevant documents to perform the method described in Section 4.2.3.

In a future version, the contextual description of the user (using roles, tasks, and preferences) will also be used, which will allow to identify other users with similar context and suggest results accordingly. This feature could be paraphrased as "users who have contexts similar to yours were also interested in documents x, y, z."

4.2.5 The keyword-context continuum

Figure 7 shows the slider that the user can use to specify the "Percentage of considered context". This is used to effectively move from the keyword-based approach to the context-based approach discussed in Section 4.2.1. This is implemented by providing a relative weight to the user-specified keywords and the context-specified keywords. This is done by considering the slider value/100 as α, the relative importance of context-specified keywords, and 1-α as the relative importance of user-specified keywords.

If the slider is in position 0%, the relative importance of context-specified keywords will be 0 and the importance of user-specified keywords will be 1. Hence, only user-specified keywords will be considered by Solr in the document retrieval process.

If the slider is in the position 100%, the relative importance of context-specified keywords will be 1 and the importance of user-specified keywords will be 0. Hence, only context-specified keywords will be considered by Solr in the document.

5 Conclusion

Intelligence analysts are faced with information and cognitive overload problems. To address these problems, it is relevant to go beyond traditional knowledge exploitation and management approaches and make use of emerging cognitive support tools. The research presented in this report proposes an integration framework to lay the groundwork for the creation of a context aware intelligence virtual analyst. Such an intelligence virtual analyst would provide essential support to human analysts faced with information and cognitive overload problems.

The results from the thorough analysis that was performed to identify the best technological and architectural candidates were presented. A target integration framework architecture was also proposed. An initial instantiation of analysis components on the proposed framework was also described.

Using this proposed integration framework and adding new analysis functionalities will allow iteratively converging towards the development of a full-fledged Intelligence Virtual Analyst Capability (iVAC).

This page intentionally left blank.

References

[1] Roy, J., and Auger, A., "The Multi-Intelligence Tools Suite—Supporting Research and Development in Information and Knowledge Exploitation", in Proceedings of 16th International Command and Control Research and Technology Symposium "Collective C2 in Multinational Civil-Military Operations", Québec City, Québec, Canada, June 21–23, 2011.

[2] Bergeron Guyard, A., and Roy, J., "Toward Case-Based Reasoning for Maritime Anomaly Detection: A Positioning Paper", in Proceedings of the Twelfth IASTED International Conference on Intelligent Systems and Control (ISC 2009), Cambridge, Massachusetts, November 2009.

[3] Bergeron Guyard, A., "Case-Based Reasoning for Maritime Anomaly Detection", in Proceedings of Cognitive systems with Interactive Sensors (COGIS 2010), Crawley, United Kingdom, November 2010.

[4] Roy, J., "Rule-Based Expert System for Maritime Anomaly Detection", in Proceedings of Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense VIX, SPIE Defense, Security, and Sensing 2010, Orlando, FL, USA, April 5–9, 2010.

[5] Roy, J., and Bergeron Guyard, A., "Supporting Threat Analysis Through Description Logic Reasoning", in Proceedings of 2012 IEEE Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), New Orleans, LA, USA, March 6–8, 2012.

[6] Poussart, D., "Future Intelligence Analysis Capability—Towards a Cohesive R&D Program Definition", DRDC – Valcartier Research Centre, Internal Draft Report, last revised in March 2013.

[7] Burkov, A., Michaud, G., and Fujitsu Consulting, "A Survey of Available Technology and Recommendations for Building an iVAC Capability", DRDC – Valcartier Research Centre, DRDC-RDDC-2014-C218, Scientific Authority: Alexandre Bergeron Guyard, August 2014.

[8] Vilus, L., and Fujitsu Consulting, "Intelligence Virtual Analysis Capability (iVAC)—Framework and Components, High-level Software Architecture Description (SAD)", DRDC – Valcartier Research Centre, DRDC-RDDC-2014-C217, August 2014.

[9] Retrieved from http://scikit-learn.org/stable/ (last accessed 09/2014).

[10] Retrieved from http://www.nltk.org/ (last accessed 09/2014).

[11] Retrieved from https://www.python.org/ (last accessed 09/2014).

[12] Retrieved from http://www.jython.org/ (last accessed 09/2014).

[13] Retrieved from http://www.ubuntu.com/ (last accessed 09/2014).

[14] Retrieved from http://www.jboss.org/ (last accessed 09/2014).

[15] Retrieved from http://activemq.apache.org/ (last accessed 09/2014).

[16] Retrieved from http://www.eclipse.org/jetty/ (last accessed 09/2014).

[17] Retrieved from http://www.oracle.com/us/technologies/java/overview/index.html (last accessed 09/2014).

[18] Retrieved from http://lucene.apache.org/solr/ (last accessed 09/2014).

[19] Retrieved from http://www.apache.org/ (last accessed 09/2014).

[20] Retrieved from http://mod-cluster.jboss.org/ (last accessed 09/2014).

[21] Retrieved from http://windows.microsoft.com/en-us/windows/home (last accessed 09/2014).

[22] Retrieved from http://msdn.microsoft.com/en-us/vstudio/aa496123.aspx (last accessed 09/2014).

[23] Retrieved from http://www.nuance.com/dragon/index.htm (last accessed 09/2014).

[24] Retrieved from http://activemq.apache.org/nms/ (last accessed 09/2014).

[25] Paquet, S., and Fujitsu Consulting, “Analysis Components Investigation Report”, DRDC – Valcartier Research Centre, DRDC-RDDC-2014-C230, Scientific Authority: Alexandre Bergeron Guyard, August 2014.

[26] Manning, C. D., Prabhakar, R. and Hinrich, S., Introduction to Information Retrieval, ISBN: 0521865719, Cambridge University Press 2008, page 178.

List of symbols/abbreviations/acronyms/initialisms

API	Application Programming Interface
DRDC	Defence Research and Development Canada
ESB	Enterprise Service Bus
GUI	Graphical User Interface
IDF	Inverse Document Frequency
ISA	Intelligent Software Assistant
ISTIP	Intelligence Science and Technology Integration Platform
iVAC	Virtual Analyst Capability
NLP	Natural Language Processing
NLU	Natural Language Understanding
OWF	Ozone Widget Framework
R&D	Research & Development
SDK	Software Development Kit
SOA	Service-Oriented Architecture
SR	Speech Recognition
SS	Speech Synthesis
TF	Term Frequency

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security markings for the title, abstract and indexing annotation must be entered when the document is Classified or Designated)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g., Centre sponsoring a contractor's report, or tasking agency, are entered in Section 8.)

DRDC – Valcartier Research Centre
Defence Research and Development Canada
2459 route de la Bravoure
Québec (Québec) G3J 1X5
Canada

2a. SECURITY MARKING (Overall security marking of the document including special supplemental markings if applicable.)

UNCLASSIFIED

2b. CONTROLLED GOODS

(NON-CONTROLLED GOODS)
DMC A
REVIEW: GCEC DECEMBER 2012

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

Integration framework for heterogeneous analysis components : Building a context aware virtual analyst

4. AUTHORS (last name, followed by initials – ranks, titles, etc., not to be used)

Bergeron-Guyard, A.

5. DATE OF PUBLICATION (Month and year of publication of document.)

November 2014

6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)

34

6b. NO. OF REFS (Total cited in document.)

26

7. DESCRIPTIVE NOTES (The category of the document, e.g., technical report, technical note or memorandum. If appropriate, enter the type of report, e.g., interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Scientific Report

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

DRDC – Valcartier Research Centre
Defence Research and Development Canada
2459 route de la Bravoure
Québec (Québec) G3J 1X5
Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC-RDDC-2014-R138

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

TIF05dz13

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

Intelligence analysts are faced with information and cognitive overload problems. To address these problems, it is relevant to go beyond traditional knowledge exploitation and management approaches and make use of emerging cognitive support tools. This report proposes an integration framework to lay the groundwork for the creation of a context aware intelligence virtual analyst. A target integration framework architecture is proposed. An initial instantiation of analysis components on the proposed framework is also described.

Les analystes du renseignement sont aux prises avec des problèmes de surcharge informationnelle et cognitive. Pour régler ces problèmes, il est essentiel d'aller au-delà des approches traditionnelles de gestion et exploitation de la connaissance et d'utiliser des outils novateurs de support cognitifs. Ce rapport propose un cadre d'intégration pour jeter les bases d'un analyste virtuel du renseignement, sensible au contexte. Une architecture cible est présentée et une première version de composantes d'analyses déployées sur le cadre est décrite.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g., Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Intelligent Software Assistant; Artificial Intelligence; Intelligence Analysis